International Journal of Control and Automation
Vol. 13, No. 4, (2020), pp. 658 - 667

Arduino Embedded Control System of DC Motor Using Proportional Integral Derivative

Iswanto[1]*, Alfian Ma'arif[2]*, Riky Dwi Puriyanto[3], Nia Maharani Raharja[4], Safinta Nurindra Rahmadhia[5]

[1]Universitas Muhammadiyah Yogyakarta, Yogyakarta, Indonesia
[2,3,5]Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[4]Universitas Islam Sunan Kalijaga, Yogyakarta, Indonesia

Email:[1]iswanto_te@umy.ac.id; [2]alfianmaarif@ee.uad.ac.id

Abstract

DC Motor has a lot of applications in the control system, robotics, industrial, and power system. The easiest and most popular control method to control DC motor is Proportional-Integral-Derivative (PID) Control. The proposed simulation has done with the great performance of the augmented system. However, simulation is an ideal situation, and most-likely is different from real-time hardware implementation. Hence, the research proposes hardware design and implementation of controlling the angular speed of the DC motor in Arduino Uno as its embedded processor system, using a PID Controller. Some examinations and analysis are done in the research, such as open-loop test, step-response, and the effect of PID parameters and sample time to the system performance. The PID controller is successfully implemented to Arduino UNO and able to control the angular speed of the DC motor. System performances differ according to the choice of PID parameters and sample time. The best PID parameters are Kp=0.7, Ki=0.3, and Kd=0.2 in 50ms sample time, as system response has no overshoot, no undershoot, fast rise, and settling time.

Keywords:*DC Motor, PID Control, Arduino Uno, Angular Speed Control, Encoder Sensor, Proportional integral derivative*

1. Introduction

Direct Current Motor is a device that converts the DC electrical energy to mechanic energy[1]. DC Motor has a lot of applications in the control system[2], robotics[3], and industrial[4]. Some examples of implementations are rocket system [5], line follower [6], line maze solving robot [7], quadrotor [8], firefighter robot [9] etc. The DC motor is very popular because it is easy to study, to control, have a good response, easy to simulate, and to make the hardware installation.

DC motor system must be able to follow a given reference value and to be stable. This problem can be overcome by some methods such as Proportional Integral Derivative (PID) Control [10], Fuzzy Logic Controller (FLC)[11], State Feedback[12], or Neural Network Control[13]. In the simulation, the problem can be solved with great results and performance[14][15]. However, a simulation is in 'ideal' condition so that real implementation in hardware may result in different performance since there is a lot of affecting factor in hardware implementation design[16].

The research proposes the hardware design of the DC motor with a low-cost embedded system device, which is Arduino Uno[17][18][19]. PID Controller is implemented to the system as its angular speed controller. PID control is chosen since it has advantages on its characteristics: easy to understand, good performance in system response, and easy to be implemented either in software simulation or hardware implementation [20].

ISSN: 2005-4297 IJCA

The structure of the paper is described as follows. The first section is the introduction. The second part is the research method that contains hardware system design, angular speed meter and PID controller. The third part is the result and discussions. The last part is the conclusion.

2. Research Method

The diagram block of an embedded system to control the speed of DC motor is shown in Figure 1. It consists of an input device, a processor, an output device, and an interface device. The input device is the encoder sensor. The output device is Driver motor L298 and DC Motor. The processor is the Arduino Uno. The interface device is a serial monitor or serial plotter from Arduino IDE.

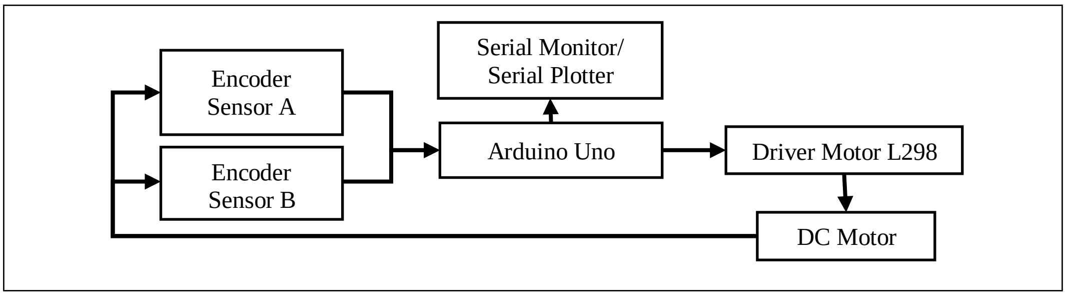

Figure 1. System BlockDiagram

The control system diagram block is shown in Figure 2. The system is categorized into a closed-loop control system. The setpoint is the reference value that must be followed by the system. PID Controller is Proportional-Integral-Derivative Control. The system to be controlled is the DC motor. The output of the augmented system is the angular speed. The feedback uses an encoder to calculate the 'real' angular speed of the system, resulted from a controlled system.

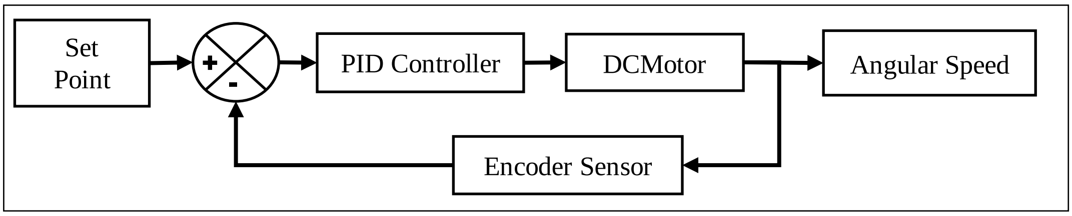

Figure 2. Control System Block Diagram

The angular speed can be obtained from calculating the pulse from the encoder in one minute. The angular speed calculates by using two encoders installed at the end of the DC motor. The calculation of angular speed can be written as

$$\omega = \frac{r}{t} \qquad (1)$$

Where r is the number of rotations in t (sample time). The number of rotations can be achieved from,

$$r = \frac{p}{p_R} \qquad (2)$$

Where p is the number of pulses in sample time, p_R is the number of pulses in one rotation. According to motor datasheet, there are 600 pulses in one rotation. Meanwhile, the sample time used in the program t_s, which is 50ms, needs to be converted with Equation below to obtain t in a minute,

$$t = \frac{t_s}{1000 * 60} \tag{3}$$

The constant 1000 is the conversion from milliseconds to seconds, and the 60 is the conversion constant from second to a minute. Thus, the RPM can be calculated as,

$$\omega = \frac{p}{600} \frac{1000 * 60}{50} = 2p \tag{4}$$

PID Control consists of proportional, integral, and derivative control. The equation of PID Control in the time domain is,

$$u(t) = K_P e(t) + \frac{K_p}{T_i} \int_0^t e(t)\, dt + K_p T_d \frac{de(t)}{dt} \tag{5}$$

Where, $u(t)$ is the control signal, K_p is the proportional control, T_i is the time of integral and T_d is the derivative time, $e(t)$ is the error or difference between the reference value and the feedback value. The PID controller alternatively can be written as

$$u(t) = K_P e(t) + K_i \int_0^t e(t)\, dt + K_d \frac{de(t)}{dt} \tag{6}$$

where

$$K_i = \frac{K_p}{T_i} \quad K_d = K_p T_d \tag{7}$$

The gain constants K_i is the integral control and K_d is the derivative control. PID control has a characteristic that affects the system response. It is because of the different structure of the controller. The proportional control corresponds to the error. The integral control corresponds to the summing error. The derivative control corresponds to the delta error.

3. Result and Analysis

In the section, there are some examinations as follows. The first part is about the open-loop system response. The second part is about the step system response of the PID Controller. The third is about various setpoint response and the effect of varying sample time.

4.1. Open Loop System Response

The result of the open-loop test hardware implementation is shown in Table 1 and Figure 3. The Arduino Uno uses an 8-bit PWM, with the range is between 0-255. The driver, motor power supply, is 12volt, and the maximum voltage to DC motor is 10volt. The minimum PWM is 50, and if the PMW is below 50, the DC motor cannot be rotated. There is two sample time used in the research, 50ms, and 100ms.

Table 1. The relation between the voltage and angular speed

PWM (8-bit)	Voltage (volt)	RPM Motor Specification	RPM Using 50ms sample time	RPM Using 100ms sample time
50	2.8	98	105	108
75	5.0	175	190	197
100	6.5	227	250	257
125	7.6	266	290	294
150	8.1	283	316	319
175	8.6	301	337	338
200	9.0	315	352	351
225	9.4	329	365	361
250	9.6	336	376	374

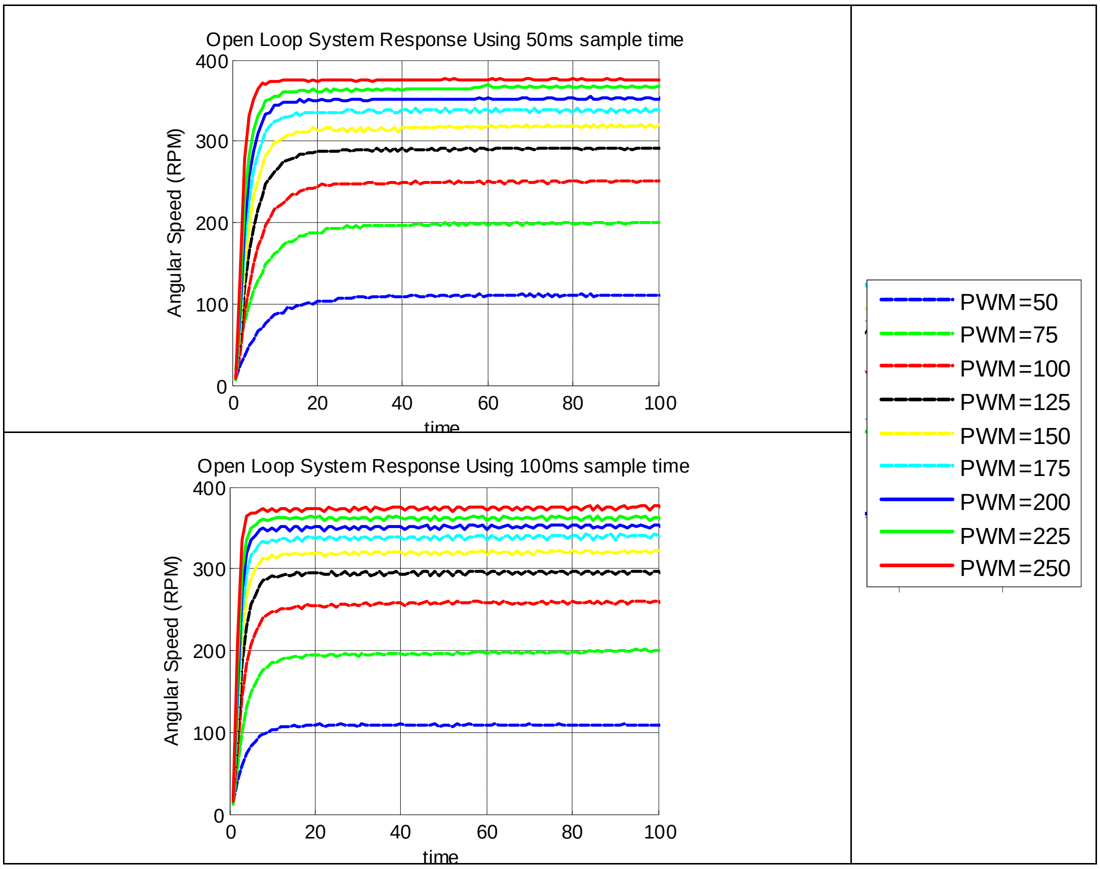

Figure 3. Open-Loop System Response Using 50ms and 100ms Sample Time

Based on Figure 3, the 50ms sample time provides a more stable angular speed than 100ms sample time. It can be seen clearly in system response with PWM=250 that 100ms sample time results in a response with some oscillations (some voltage ripples) after reaching a steady state. Sample time is essential to the accuracy and speed response of the system. By using smaller sample time, for example, 50ms, the delay for the system to respond to error is also smaller. Hence, the system is able to prevent the output from making any error before the error could happen. The augmented system then will respond in faster response and result in a more stable output.

4.2. Step Response of DC Motor with PID Controller

The proportional control system response is shown in Table 2 and Figure 4. The setpoint is 100RPM. In Table 2, the increasing proportional control can reduce the steady-state error. In Figure 4, it can be seen that the steady-state error is reduced. Proportional control increases the overshoot and rise time. Thus, the hardware implementation can be seen that proportional control affects reducing rise time, increasing overshoot, and reducing the steady-state error.

Table 2. The system response of Proportional Control

Kp	Ki	Kd	Rise Time	Settling Time	Overshoot	Steady State Error
0.5	0	0	-	-	-	64
0.75	0	0	-	-	-	48
1	0	0	-	-	-	38
1.25	0	0	1.7727	-	-	32
1.5	0	0	1.4286	-	12	26

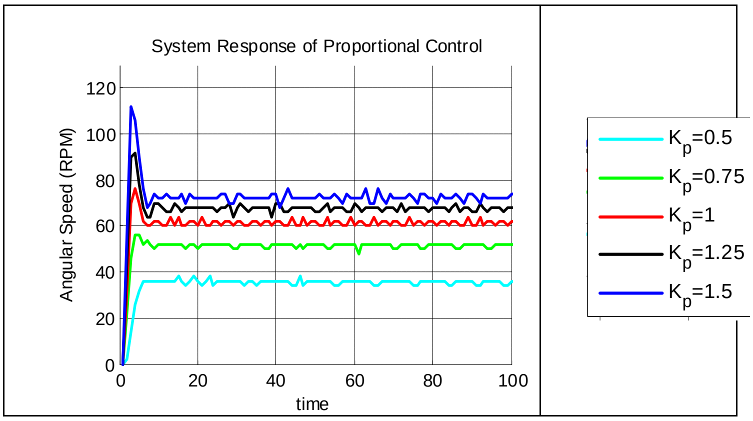

Figure 4. Closed-Loop System Response of Proportional Control

The Integral control system response is shown in Table 3 and Figure 5. It can be seen that increasing the integral control can eliminate the steady-state error and result in a more aggressive system response. The bigger integral control has a faster rise time, but it has bigger overshoots and undershoots. The integral control may not be too big, or it can make the system have big overshoots and undershoot. Thus, the integral control affects increasing overshoot, increasing undershoot, reducing rise time, and eliminating the steady-state error.

Table 3. The system response of Integral Control

Kp	Ki	Kd	Rise Time	Settling Time	Overshoot	Steady State Error
0.5	0	0	-	-	-	64
0.5	0.1	0	9.1667	18	2	0
0.5	0.2	0	3.2321	11.5	6	0
0.5	0.3	0	2.2895	56.5	20	0
0.5	0.4	0	2.0317	34.5	28	2

ISSN: 2005-4297 IJCA

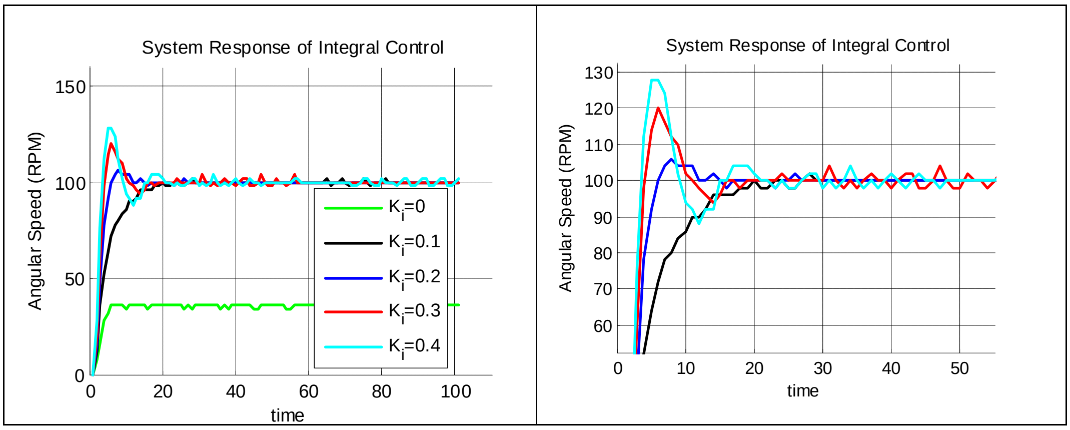

Figure 5. Closed-Loop System Response of Integral Control

The Derivative Control system response is shown in Table 4 and Figure 6. It can be seen that increasing the derivative control can reduce the overshoot but make the system have the undershoot. The bigger the derivative control, the bigger the undershoot is. The derivative control can increase the rise time, but it also can reduce the rise time after the undershoot appears. Thus, the derivative control effects for reducing the overshoot, reducing the rise time, increasing the undershoot.

Table 4. The system response of Derivative Control

Kp	Ki	Kd	Rise Time	Settling Time	Overshoot	Steady State Error
0.6	0.3	0	2.2857	60.5	16	2
0.6	0.3	0.1	2.3875	10.5	8	0
0.6	0.3	0.2	2.2778	11.5	6	2
0.6	0.3	0.3	2	13.3333	6	0
0.6	0.3	0.4	1.7311	97.3333	4	0

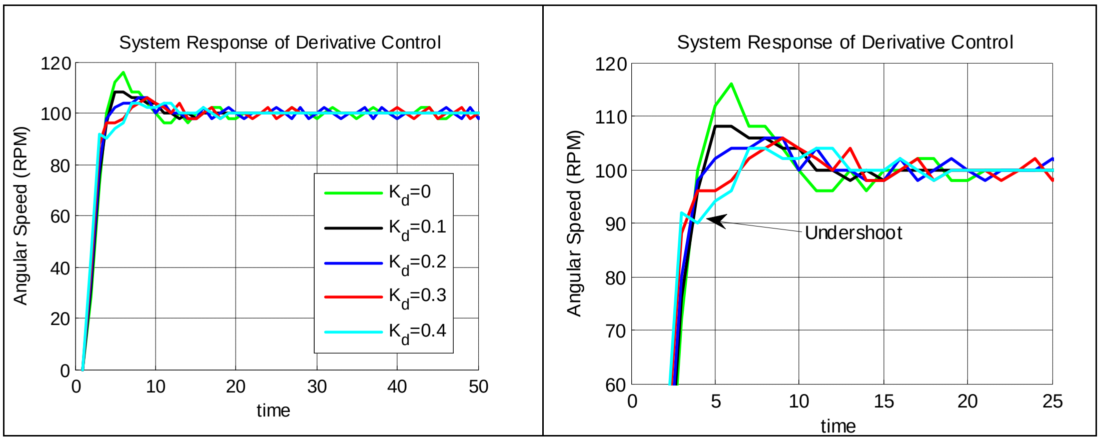

Figure 6. Closed-Loop System Response of Derivative Control

Table 5 is the summary of the PID controller characteristic based on the hardware system response implementation. The proportional control and integral control is suitable for reducing the rise time but have a risk of increasing the overshoot. The best function of proportional control is reducing the rise time, and the best function of integral control is eliminating the steady-state error. The derivative control is

ISSN: 2005-4297 IJCA

suitable for reducing the overshoot but has a risk of increasing the undershoot. Thus, derivative control cannot be given too big.

Table 5. The system response of Parameter PID controller

	Rise Time	Settling Time	Overshoot	Undershoot	Steady State Error
Proportional Control	Reducing	Inconsistent	Increasing	Insignificant	Reducing
Integral Control	Reducing	Inconsistent	Increasing	Insignificant	Eliminating
Derivative Control	Inconsistent	Inconsistent	Reducing	Increasing	Insignificant

4.3. Set Point and Sample Time Response

The best PID parameter is shown in Table 6 and Figure 7. The setpoint is 100RPM. The best PID control parameter is number 4. The system response number 2 and number 3 have an undershoot. It is because of the big derivative control. The increasing derivative control must be careful because it will give the undershoot response. The overshoot response gives by number 1 because of the big proportional value.

Table 6. The system response of Parameter PID controller

No	Kp	Ki	Kd	Rise Time	Settling Time	Overshoot	Steady State Error
1	0.8	0.3	0.1	1.7115	88.3333	8	0
2	0.75	0.3	0.3	1.6317	86.3333	4	0
3	0.75	0.3	0.25	1.6573	66.3333	4	2
4	0.7	0.3	0.2	1.9167	14.5	2	2

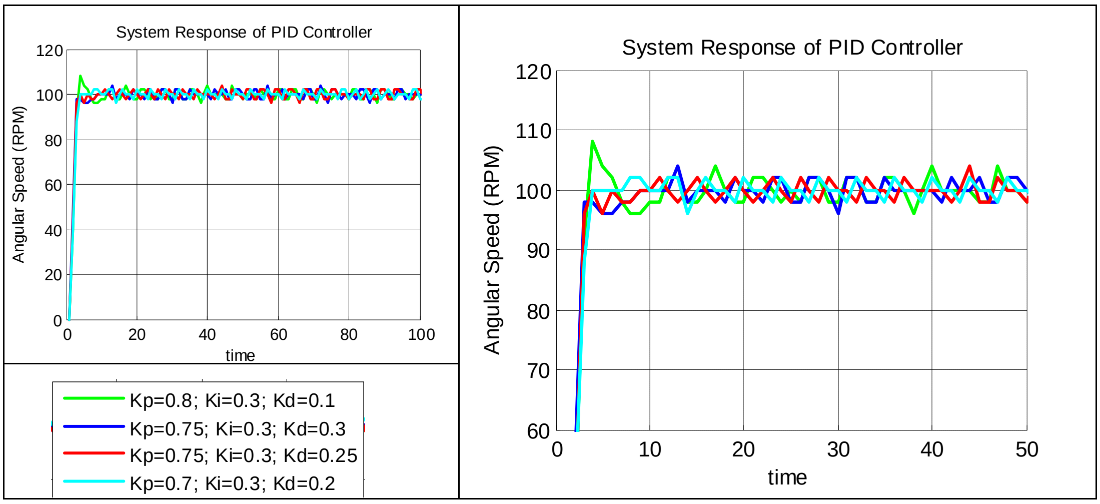

Figure 7. Closed-Loop System Response of PID Controller

ISSN: 2005-4297 IJCA

The next examination is about setpoint and sample time response. There is two sample time used in the research, 50ms, and 100ms. The result is shown in Table 7 and Figure 8. The PID controller can control and stabilize the system in some set points and reach the reference signal. The sample time affects the system response but still can follow the set point. The 50ms sample time gives faster response than the 100ms response. Thus the smaller sample time is good for faster system response. But, it cannot be too small because it can eliminate the original characteristics of angular speed data.

Table 7. The system response various set point and sample time

Kp	Ki	Kd	Sample Time = 50ms				Sample Time = 100ms			
			Rise Time	Settling Time	Over-shoot	Steady State Error	Rise Time	Settling Time	Over-shoot	Steady State Error
0.7	0.3	0.2	3.25	-	4	2	5.27	73.5	4	0
0.7	0.3	0.2	1.74	67.33	4	0	1.25	10	20	2
0.7	0.3	0.2	1.64	6.75	6.67	0	1.12	11.33	26.67	0
0.7	0.3	0.2	1.45	5.5	9	0	0.86	11.57	32	1
0.7	0.3	0.2	1.61	7.25	2.4	0	1.04	10	20	2
0.7	0.3	0.2	2.09	12	3.33	2	1.42	7.5	5.67	2

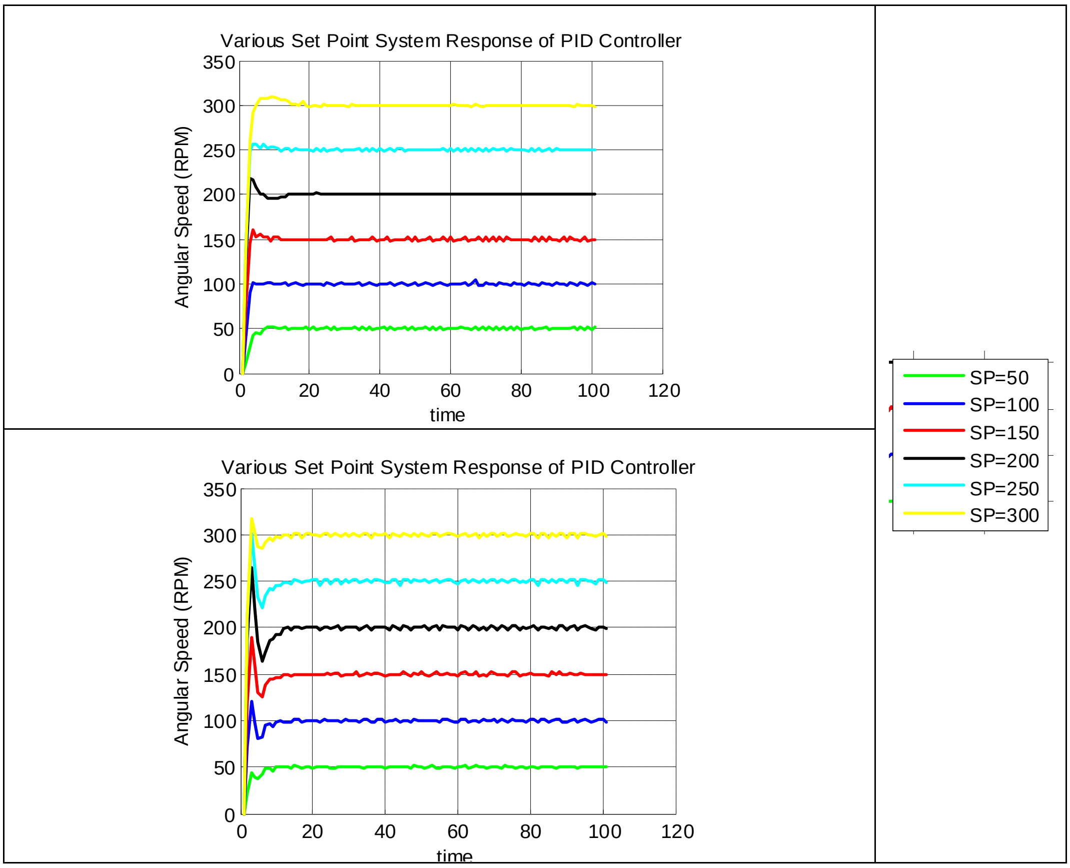

Figure 8. Various and Sample Time Response of Closed Loop System Response of PID Controller

ISSN: 2005-4297 IJCA

5. Conclusions

The research is control of the DC motor system using the Proportional Integral Derivative (PID) Control using embedded system Arduino Uno. The PID controller can control and stabilize the DC motor in the Embedded System using Arduino Uno. It can reach some various set points with settling time below one second. The proportional control has characteristics for reducing the rise time but increasing the overshoot. The integral control has characteristics to eliminate the steady-state error and increase the overshoot. The derivative control has characteristics to reduce the overshoot but increasing the undershoot. Smaller sample time provides faster and more stable system response. However, sample time could not be too short of providing the original characteristics of the output. The best PID controller for the 100RPM set point is $Kp = 0.7, Ki = 0.3, Kd = 0.2$ with 50ms sample time.

Acknowledgments

The authors would like to thank the editor and reviewer for the advice, suggestion and comment to improve the quality of the paper. Especially thank the editor that helps the editorial process from the submission process until the manuscript is published.

References

[1] M. R. Qader, "Identifying the Optimal Controller Strategy for DC Motors," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 6, no. 4, p. 252, 2017.
[2] M. D. Dzulfiqar and N. S. Widodo, "Implementasi PID Navigasi Pelacakan Titik Api dengan Sensor Flame Array pada Robot Hexapod untuk Kontes Robot Pemadam Api Indonesia," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 1, no. 3, p. 131, 2019.
[3] M. Kamaludin and W. S. Aji, "Manuver Robot Manual Menggunakan PID pada Robot Manual KRAI 2018," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 1, no. 3, p. 91, 2019.
[4] R. D. P. Fahmi Abdul Aziz, "Rancang Bangun Mesin Pengecat Dinding Otomatis Berbasis PLC CP1E-NA20DR-A," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 1, no. 3, pp. 118–130, 2019.
[5] D. Lastomo, H. Setiadi, and M. R. Djalal, "Optimization pitch angle controller of rocket system using improved differential evolution algorithm," *International Journal of Advances in Intelligent Informatics*, vol. 3, no. 1, pp. 27–34, 2017.
[6] A. Latif, H. A. Widodo, R. Rahim, and K. Kunal, "Implementation of Line Follower Robot based Microcontroller ATMega32A," vol. 1, no. 2, pp. 70–74, 2020.
[7] A. Maarif, S. Iskandar, and I. Iswanto, "New Design of Line Maze Solving Robot with Speed Controller and Short Path Finder Algorithm," *International Review of Automatic Control (IREACO)*, vol. 12, no. 3, p. 154, 2019.
[8] I. Iswanto, A. Ma'arif, O. Wahyunggoro, and A. Imam, "Artificial Potential Field Algorithm Implementation for Quadrotor Path Planning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 8, 2019.
[9] A. Latif, K. Shankar, and P. T. Nguyen, "Legged Fire Fighter Robot Movement Using PID," *Journal of Robotics and Control (JRC)*, vol. 1, no. 1, pp. 15–18, 2020.
[10] A. Ghareaghaji, "A Comparison between Fuzzy-PSO Controller and PID-PSO Controller for Controlling a DC Motor," *Bulletin of Electrical Engineering and Informatics*, vol. 4, no. 2, pp. 130–135, 2015.
[11] M. S. Mohiuddin, "Performance Comparison of Conventional Controller with Fuzzy Logic Controller using Chopper Circuit and Fuzzy Tuned PID Controller," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 2, no. 4, pp. 189–200, 2014.
[12] M. Ahmad, A. Khan, M. A. Raza, and S. Ullah, "A study of state feedback controllers for pole placement," in *5th International Multi-Topic ICT Conference: Technologies For Future Generations, IMTIC 2018 - Proceedings*, 2018.
[13] Y. Naung, S. Anatolii, and Y. Htet Lin, "Speed control of DC motor by using neural network parameter tuner for PI-controller," *Proceedings of the 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2019*, no. Ci, pp. 2152–2156, 2019.
[14] S. J. Hammoodi, K. S. Flayyih, and A. R. Hamad, "Design and implementation speed control system of DC motor based on PID control and matlab simulink," *International Journal of Power Electronics and Drive Systems*, vol. 11, no. 1, pp. 127–134, 2020.
[15] A. Ma'Arif, H. Nabila, Iswanto, and O. Wahyunggoro, "Application of Intelligent Search Algorithms in Proportional-Integral-Derivative Control of Direct-Current Motor System," in *Journal of Physics: Conference Series*, 2019, vol. 1373, no. 1.
[16] A. R. Al Tahtawi and M. Yusuf, "Low-cost quadrotor hardware design with PID control system as flight

controller," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 4, p. 1923, 2019.
[17] I. Prasojo, A. Maseleno, O. Tanane, and N. Shahu, "Design of Automatic Watering System Based on Arduino," *Journal of Robotics and Control (JRC)*, vol. 1, no. 2, pp. 59–63, Jan. 2020.
[18] P. Megantoro, A. Widjanarko, R. Rahim, K. Kunal, and A. Z. Arfianto, "The Design of Digital Liquid Density Meter Based on Arduino," *Journal of Robotics and Control (JRC)*, vol. 1, no. 1, pp. 1–6, 2020.
[19] A. Budianto, W. S. Pambudi, S. Sumari, and A. Yulianto, "PID control design for biofuel furnace using arduino," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 16, no. 6, pp. 3016–3023, 2018.
[20] A. A. T., P. J. O., O. O., and A. S. L., "PID controller for microsatellite yaw-axis attitude control system using ITAE method," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 2, p. 1001, 2020.

Authors

Iswantois a lecturer in Universitas Muhammadiyah Yogyakarta.

Alfian Ma'arifis a lecturer in Universitas Ahmad Dahlan.

Riky Dwi Puriyantois a lecturer in Universitas Ahmad Dahlan.

Nia Maharani Raharjais a lecturer in Universitas Islam Sunan Kalijaga.

Safinta Nurindra Rahmadhiais a lecturer in Universitas Ahmad Dahlan.

ISSN: 2005-4297 IJCA